# Exercise: Source Control Systems

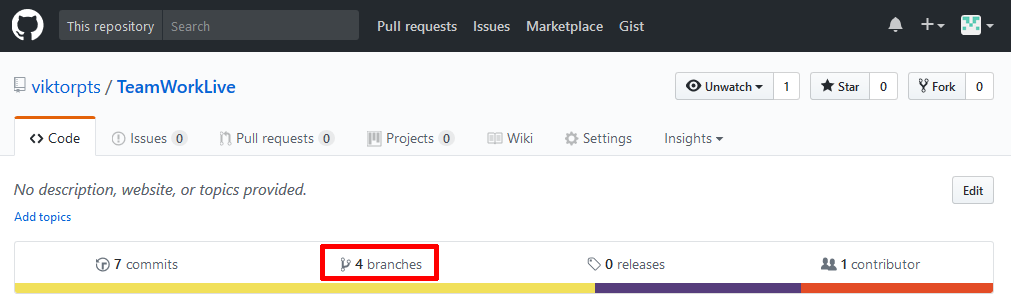
# Stage 7, Task 1 – Merge Notifications into master

**To begin, make sure Stage 4, Stage 5 and Stage 6 are fully finished, committed to their respective branches and conflicts are resolved.**

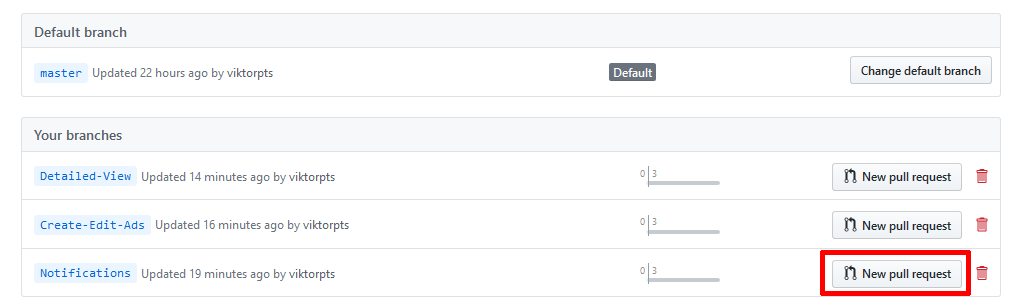
It’s likely the three features we developed in branches may conflict with each other – it’s time to pull them into the master one by one and resolve any merge conflicts that have been created.

First, we’re going to create a **pull request** for branch Notifications. A pull request signals the readiness of our branch to be merged into the production level of development (or into an upstream branch, that may later go into production). Then the manager of the project may review the changes and approve our request for code to be pulled from our branch into the master. For larger or sensitive projects, it’s a good idea to give the right to approve pulls into the master only to senior level developers, or just the project lead.

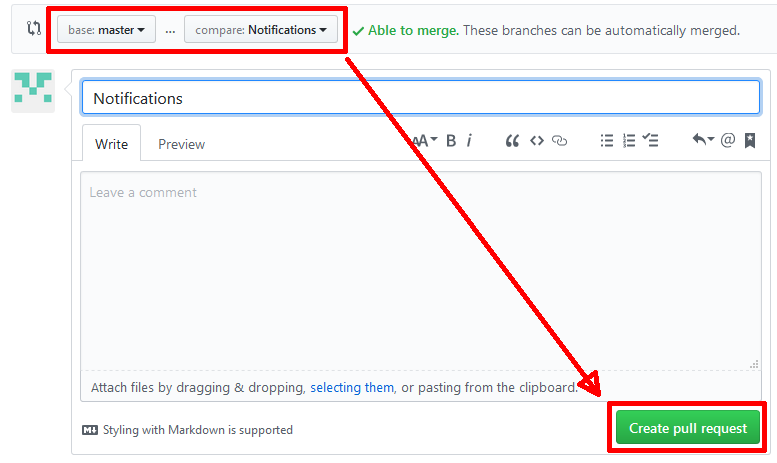
Open your project’s **GitHub page** and navigate to the **“branches”** section:



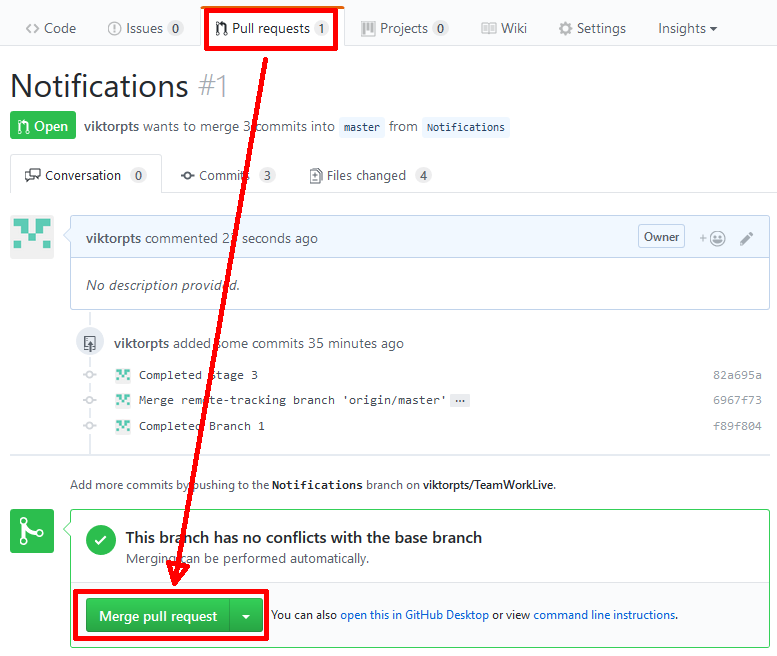
Click **“New pull request”** beside the branch **“Notifications”**:



Leave a note if you wish and make sure the target is set to “master”. The message “Able to merge” at the top means there are no conflicts and you can merge the changes automatically:



After the request is created, the page where it can be managed should open automatically. If not, you can see it from “Pull requests” in the navigation pane. If you see the green button “Merge pull request”, then congratulations, you can seamlessly merge the branch into the master without conflicts!



If not, then you must resolve the conflicts. If you’ve followed the tasks so far, **you should not encounter problems** **here**, but if you do, you need to pull the master into the Notifications branch, resolve the conflicts and then make a new pull request into the master.

Make sure you pull the master locally and test the application before proceeding with the next task!